CS320 Web and Internet Programming
JSP Action Elements

Chengyu Sun
California State University, Los Angeles

JSP Action Elements

```
<prefix:action attr1="value1" attr2="value2" …>
        action_body
</prefix:action>

<prefix:action attr1="value1" attr2="value2" …/>
```

Action Types

- JSP Standard
- JSP Standard Tag Library (JSTL)
- Custom

JSP Standard Actions

- Bean
  - <jsp:useBean>
  - <jsp:getProperty>
  - <jsp:setProperty>
- Multi-page
  - <jsp:include>
  - <jsp:forward>
  - <jsp:param>
- Attribute
  - <jsp:attribute>
  - <jsp:body>
- Document
  - <jsp:plugin>
  - <jsp:element>
  - <jsp:text>

JavaBeans

- Initially designed for GUI builders
- *Beans* support
  - Introspection
  - Customization
  - Events
  - Properties
  - Persistence

JavaBeans for Dummies …

```
public class IamABean {

        int prop1;
        String prop2;
        boolean prop3;

    public IamABean() { prop1 = 0; prop2=""; prop3=false; }

    public getProp1() { return prop1; }

    public getProp2() { return prop2; }

    public setProp2( String s ) { prop3 = new String(s); }

    public isProp3() { return prop3; }

}
```

… JavaBeans for Dummies

- A zero-argument constructor
- No public class variables
- Property xxx
  - getXxx() or isXxx()
  - setXxx()

VisitorsWithBeans.jsp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Visitors with Beans</TITLE> </HEAD>

<jsp:useBean id="visitor" class="edu.calstatela.cs.csun.Visitor" />

<BODY>Hello, you are visitor
No.<jsp:getProperty name="visitor" property="fake" />,
or actual visitor No.<jsp:getProperty name="visitor" property="real" />.

<!-- Page created on <%= new java.util.Date() %> -->
</BODY>
</HTML>
```

Bean Tags and Attributes

- jsp:useBean
  - class
  - id
  - scope
    - page (default)
    - request
    - session
    - application
- jsp:getProperty
  - name
  - property
- jsp:setProperty
  - name
  - property
  - value

Visitor.java

```
package edu.calstatela.cs.csun;

public class Visitor {

      int    real;

   public Visitor() { real = 0; }

   public int getReal() { return real++; }

   public int getFake() { return (int) (Math.random() * 1000000); }

}
```

Java Packages

- *Java does not permit a class without a package qualifier to be used in a class that belongs to a package. (Enforced in JDK 1.4)*
- Naming convention
- Package name ⇔ path

Deploy JSP with Packages on the CS Server

- JSP files under `~/public_html`
- Package directory under `~/public_html/WEB-INF/classes`
- Access `Whatever.jsp` as `http://cs.calstatela.edu:8280/~username/Whatever.jsp`

Example of setProperty

```
<jsp:setProperty name="fb" property="cls" value="cs320"/>

<jsp:setProperty name="fb" property="fname"
  value='<%= request.getParameter("fname")%>'/>

<jsp:setProperty name="fb" property="age">
  <jsp:attribute name="value" trim="true">
    <%= request.getParameter("age")%>
  </jsp:attribute>
</jsp:setProperty>

<jsp:setProperty name="fb" property="lname"/>
<jsp:setProperty name="fb" property="*"/>
```

Tag Libraries

- JSTL
- Custom Tag Libraries
  - Java classes
  - Tag files

VisitorsTL.jsp

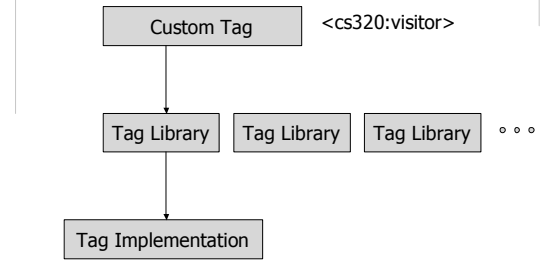
```
<%@ taglib prefix="cs320" uri="http://www.calstatela.edu/csun/cs320" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Visitors with Tag Library</TITLE> </HEAD>

<BODY>Hello, you are visitor No.<cs320:visitor fake="true"/>,
or actual visitor No.<cs320:visitor fake="false"/>.

<!-- Page created on <%= new java.util.Date() %> -->
</BODY>
</HTML>
```

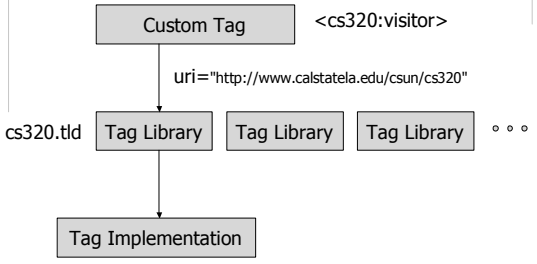
From Custom Tags to Their Implementations



taglib Directive

- `prefix` – library name used in this page
- `uri` – library identifier
  - Must be globally unique
  - Must match the library identifier in the TLD
  - Convention – URL-like

From Custom Tags to Their Implementations

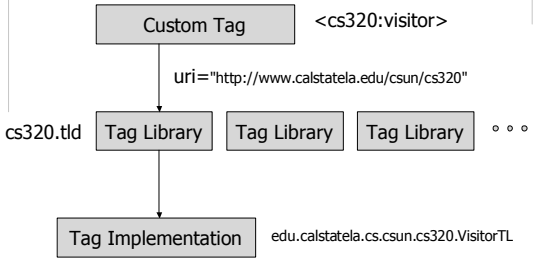


cs320.tld

```
<?xml version="1.0" ?>

<taglib>
   <uri>http://www.calstatela.edu/csun/cs320</uri>
   <tlibversion>1.0</tlibversion>
   <jspversion>2.0</jspversion>
   <shortname>CS320 Custom Tag Samples</shortname>
   <info> Custom tag samples for CS320</info>
   <tag>
      <name>visitor</name>
      <tagclass>edu.calstatela.cs.csun.cs320.VisitorTL</tagclass>
      <bodycontent>empty</bodycontent>
      <attribute>
         <name>fake</name>
         <required>true</required>
      </attribute>
   </tag>
</taglib>
```

From Custom Tags to Their Implementations



VisitorTL.java …

```
package edu.calstatela.cs.csun.cs320;

import java.io.*;
import javax.servlet.jsp.*;
import javax.servlet.jsp.tagext.*;

public class VisitorTL extends SimpleTagSupport {

      static int     actualVisitor = 0;

      // attributes
      boolean fake;

   public VisitorTL() { fake = false; }
```

… VisitorTL.java

```
   public void setFake( String fake )
   {
      this.fake = Boolean.valueOf(fake).booleanValue();
   }

   public void doTag() throws IOException
   {
      JspWriter out = getJspContext().getOut();
      out.print( fake ? (int) (Math.random()*1000000) : actualVisitor++ );
   }

} // end of class VisitorTL.java
```

Deploy a Custom Tag Library

- Class files
  - src/edu/calstatela/cs/csun/cs320/*.class
- TLD file
  - src/META-INF/cs320.tld
- Create jar file under src/
  - jar -cvf cs320.jar edu META-INF
- cs320.jar goes to
  - cs server: ~/public_html/WEB-INF/lib
  - On your home machine: $(TOMCAT)/webapp/ROOT/WEB-INF/lib

JSP Standard Tag Library (JSTL)

Library	URI	Prefix
Core	http://java.sun.com/jsp/jstl/core	c
XML Processing	http://java.sun.com/jsp/jstl/xml	x
I18N Formatting	http://java.sun.com/jsp/jstl/fmt	fmt
Database Access	http://java.sun.com/jsp/jstl/sql	sql
Functions	http://java.sun.com/jsp/jstl/functions	fn

http://java.sun.com/products/jsp/jstl/1.1/docs/tlddocs/index.html

JSTL Examples

- jstl1.jsp
- jstl2.jsp
  - <c:out value="${1+2+3}"/>

Setting Action Attribute Values

- JSP Expression
- <jsp:attribute>
- JSP Expression Language (EL)

```
<jsp:setProperty name="fb" property="fname"
 value='<%= request.getParameter("fname")%>'/>

<jsp:setProperty name="fb" property="age">
 <jsp:attribute name="value" trim="true">
  <%= request.getParameter("age")%>
 </jsp:attribute>
</jsp:setProperty>

<c:out value="${1+2+3}"/>
```

EL Opertors and Implicit Variables

- +,-,*,/,%
- ==, !=, <, >, <=, >=
- eq, ne, lt, gt, le, ge
- &&, ||, !
- and, or, not
- ?:
- .
- []
- empty

- param
- paramValues
- header
- headerValues
- cookie
- …

JSTL Core

- <c:out>
  - value
- <c:forEach>
  - items
  - var
  - begin, end, step
- <c:forToken>
  - items, delims
  - var
  - begin,end, step

- <c:if>
  - test
  - var
  - scope
- <c:choose>
- <c:when>
  - test
- <c:otherwise>

JSTL Functions

- fn:length()
- fn:contains()
- fn:startWith()
- fn:endsWith()
- fn:indexOf()
- fn:replace()
- fn:split()
- fn:toUpperCase()
- fn:toLowerCase()
- fn:substring()
- fn:substringAfter()
- fn:substringBefore()
- fn:trim()